**Operator定义**

COStream流语言中最基本的组成单元和计算单元是operator，即在数据流图中表示一个计算结点。operator定义了actor输入、输出流和具体的计算处理过程。如图1所示，operator由头部定义和体定义组成，其中operator都定义了处理的输入和输出流，operator体包括变量声明和定义、init、work和window四个部分，其中work是必须的，其部分是可选的。



图1：operator的构成

类似于C语言的函数文法，每一个operator均由一个头部和主体部分组成，其BNF可表示为：*opInvoke ::=opInvokeHeadop InvokeBody*

**operator头部定义**

头部opInvokeHead定义了operator的名称和operator用于计算和通信的输入、输出流，BNF表示为：

*opInvokeHead ::= opOutputs ‘=’ ID opInputs | ID opInputs*

*opOutputs ::=streamID |‘(’ streamID+, ‘)’*

*opInputs ::= streamID |‘(’ streamID +, ‘)’*

*streamID ::= streamType ID | ID*

以图1.1的Averager为例，其输入流为S，输出流为P，完整头部表示为：

*P = Averager(S){ // opInvokeHead*

*…// opInvokeBody*

*}*

**多输入多输出operator支持**

与StreamIt不同的是，除了单输入、单输出流外，COStream的operator也支持流的多输入，多输出，具体用法如下：

*(O1,O2,O3,O4,O5) = MyOperator(IN1,IN2,IN3,IN4,IN5)*

*{*

*…*

*}*

多个输出流必须分开定义，并用“( )”括起；同理多个输入流也分开定义。

**operator主体定义**

operator的主体部分包含在“{”和“}”中，定义了所代表的operator的具体运算过程，它的BNF表示如下：

*opInvokeBody ::= '{'*

*( declaration+ ) \**

*( ‘init’ opInvokeInitWork+) ?*

*( ‘work’ opInvokeInitWork+)*

*( ‘window’ opInvokeWindow+) ?*

*'}'*

*opInvokeInitWork ::= stmt*

*opInvokeWindow ::= '{'*

opInvokeWindowBody? ;

*'}'*

*opInvokeWindowBody ::= ID ‘sliding’'(' (expr ‘,‘ expr)? ')'*

*| ID ‘tumbling''(' expr? ')'*

operator体包括变量的声明和定义、init、work和window四个部分，其中work是必须的，其部分是可选的。